Development and Evaluation of a Retrieval-Augmented Generation (RAG) System for Clinical Guidelines

Author Name
Institut für interventionelle und diagnostische Radiologie der TUM

Abstract

LLMs have shown great potential to streamline workflows, especially in natural language-based tasks. This might make them useful for combing through guidelines for clinicians seeking rapid and context-specific answers. However, LLMs' ability to recall accurate information is still limited, and they are prone to retrieving incomplete or hallucinated information. This presents a challenge for their use in medical fields, where precise and correct information is paramount for adequate patient care. This study presents the development and evaluation of a Retrieval-Augmented Generation (RAG) system using entirely pre-trained models, based on the ESUR guidelines, aiming to improve accessibility and usability. Our evaluation demonstrates that the RAG system effectively handles the intricacies of the ESUR guidelines, achieving a retrieval accuracy of X% and a response relevance score of Y based on expert evaluation. This study shows an example of the potential of RAG systems in supporting clinical decision-making while maintaining a high level of accuracy.

1 Introduction

Time constitutes a critical resource in clinical practice, as delays in decision-making can result in unfavorable patient outcomes and increased costs for healthcare systems. Clinicians often rely on detailed guidelines—such as those provided by the European Society of Urogenital Radiology (ESUR)—to inform diagnostic and therapeutic choices. These guidelines, however, can exceed hundreds of pages and are frequently updated, making it challenging for clinicians to locate specific information in a fast-paced environment. Rapid and accurate access to these resources could therefore greatly enhance both the efficiency and quality of patient care. Recent advances in large language models (LLMs) have demonstrated exceptional capabilities in processing and generating domain-specific information, offering the prospect of streamlined information retrieval [1]. Unlike manual lookups—which demand substantial cognitive effort and time—LLMs provide near-instantaneous access to extensive, structured knowledge[2]. This speed may prove invaluable in high-pressure clinical contexts where every second is critical. Nonetheless, these models exhibit a significant shortcoming: unreliability. They can produce factually incorrect or fabricated ("hallucinated") outputs[3], and their dependence on pre-trained knowledge hampers the incorporation of newly issued or updated guidelines without additional training. In clinical settings, where precision is paramount, such limitations pose serious risks[4]. Retrieval-augmented generation (RAG) systems have been proposed as promising solutions to combine the rapidity of LLMs with the level of reliability demanded in clinical decision-making [5], [6], [7]. By supplementing LLM outputs

with relevant, up-to-date data from external sources, RAG systems maintain the generative speed of LLMs while grounding responses in authoritative materials. This approach has been shown to substantially mitigate hallucinations and the use of outdated information[8]. Previous work has shown that RAG systems can outperform standalone LLMs in tasks where factual accuracy and contextual relevance are crucial—attributes especially pertinent to specialized fields such as radiology[6]. Here, we present the development and deployment of a RAG system tailored for the ESUR guidelines. We detail the core components involved, including text extraction, chunking strategies, the selection of embedding models, and the optimization of retrieval processes. We also compare the system's performance—assessed by retrieval accuracy, factual correctness, completeness, and helpfulness—to a baseline LLM without retrieval augmentation. Our findings aim to underscore the potential of RAG systems to enhance the usability and accessibility of clinical guidelines, thereby offering a more efficient and reliable alternative to traditional information retrieval methods.

2 Method

2.1 Background

RAG systems were first proposed in 2020[9] to mitigate key drawbacks of standalone LLMs. The core framework, which comprises indexing, retrieval, augmentation, and generation, remains largely unchanged. Innovations and refinements have emerged in each stage, but the fundamental sequence remained unchanged[10].

2.1.1 Indexing

The process starts with selecting the textual information to augment the LLM's capabilities. This information is broken into manageable chunks, allowing the model to handle materials within its context window and avoid redundancy.Various chunking strategies have been proposed [11] (fixed-size, semantic were considered). Finally, each chunk is encoded via an embedding model, which generates high-dimensional vector representations stored in a database designed for efficient similarity-based retrieval (e.g., FAISS).

2.1.2 Retrieval

Next, the user queries are encoded using the same embedding model and vector similarity (e.g., cosine similarity) is calculated against the precomputed chunk vectors. The top k most relevant chunks are returned. More advanced methods such as example query expansion, have been shown to substantially enhance the performance of the retrieval process[12].

2.1.3 Augmentation

The retrieved chunks then serve as context for the LLM's generative process. They are fed into the model prompt alongside the user's question. This strategy has been shown to produce more precise and varied answers, mitigating the limitations of parametric-only baselines[9].

2.1.4 Generation

Finally, the LLM generates an output based on the query and the provided information.

There are additional, post processing strategies to increase accuracy, such as implementation of a recursive retrieval process to improve the quality of the output [13].

The complete workflow is illustrated in Figure 1.

2.2 System Overview

The RAG-pipeline was built in Python. The initial step involved extracting the ESUR guidelines from a PDF document into a machine-readable Text-format. Extraction into a machine-readable text file was partly automated, followed by manual cleanup to remove unnecessary elements (e.g., headers, footers), to improve the efficiency of indexing and retrieval, and the text was rearranged in accordance with the original guidelines.

The resulting Text-file was then segmented into chunks. Several chunking strategies were tested and

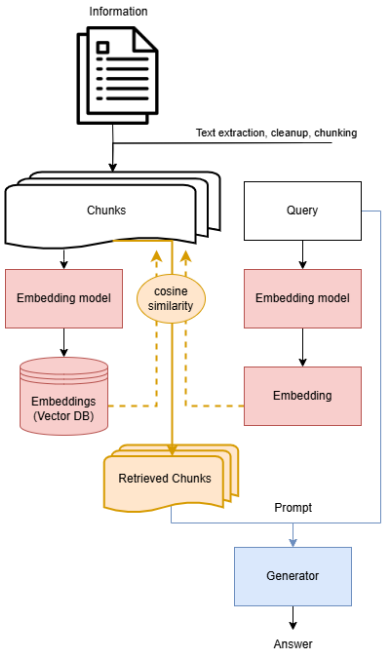

Figure 1: Basic RAG-System. Indexing is coded in red, Retrieval is coded in yellow and Augmentation/Generation in blue

manually evaluated (including fixed-window splitting with and without overlap, and semantic segmentation). Ultimately, the system adopted a headline-based chunking approach, as it efficiently captured cohesive topic blocks within the guidelines.

We evaluated multiple pre-trained models for embedding, spanning general-purpose sentence embedders (e.g., `all-MiniLM-L6-v2`, `all-mpnet-base-v2`), domain-specific biomedical/scientific models (e.g., `UMLSBert_ENG`), and specialized semantic similarity architectures (e.g., `stsb-roberta-large`). Based on a pilot set of radiology-related questions, `all-mpnet-base-v2` demonstrated superior retrieval of relevant ESUR passages. Consequently, the final system encodes each chunk into a 768-dimensional vector, storing these representations in FAISS.

In the retrieval step, the system retrieved the top k chunks. Sliding windows with and without overlap and various values of k were experimented with. Within this study and given the guideline's length and structure, we found that $k = 3$ was sufficient, since most inquiries referred to content found in a single chunk.

In the augmentation step, the retrieved passages were combined to form context that was then provided to the language model. Our model of choice for generation was a large, pre-trained architecture (`meta-llama/Meta-Llama-3-70B-Instruct-Turbo`), primed with instructions to answer questions based strictly on ESUR material and, where possible, to cite specific sections or pages. The input to the model consisted of:

- Instructions to the model: "You are an assistant that provides information based on the ESUR guidelines. Provide clear and concise answers, citing chapters or pages when possible."
- The user query.
- The context retrieved from the index.
- A brief conversation history to ensure coherence across interactions.

As illustrated in Figure 2, an extended pipeline was also explored, to see whether it was possible to enhance the quality of the output, by having the generator model rate the original output. The model was asked to evaluate whether the retrieved chunks fully answered the query, and if not, conduct a keyword expansion, query reformulation and new chunk retrieval. In any case, the model was asked to reformulate the answer, and instructed to incorporate verbatim citations only from relevant passages. This iterative process greatly increased the runtime of the system, so only a single iteration was performed.

For the comparative analysis, test queries were fed to both the "unenhanced" version of the system (a single retrieval and generation pass) and the "enhanced" iterative version. We additionally contrasted out system against a state of the art, non-RAG baseline (GPT-4o) to measure how retrieving ESUR guideline text impacted the completeness and factual correctness of generated responses.

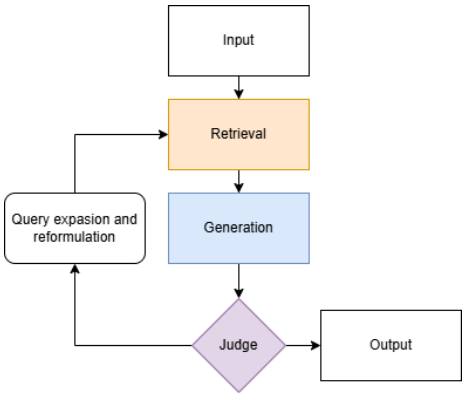

Figure 2: Enhanced RAG-Pipeline

3 Evaluation Methodology

As stated before, the relevant information for most queries typically resided in a single chunk. Therefore, retrieval metrics like precision/recall across multiple chunks were less appropriate under a fixed k. We chose to use a simple binary retrieval accuracy: the retrieval was counted as correct if the relevant chunk was included among the top k results, and incorrect otherwise.

$$\text{Accuracy} = \begin{cases} 1, & \text{if the relevant chunk is among the top-}k\text{ retrieved chunks} \\ 0, & \text{otherwise} \end{cases}$$

The overall retrieval accuracy was computed as the ratio of successful queries to the total number of queries in the test set.

$$\text{Overall accuracy} = \frac{\text{Successful retrievals}}{\text{Total queries}} \quad (1)$$

A pilot study involving 3 radiologists with between 2.5 to X years of experience was performed with manual searches of the ESUR guidelines for 13 questions that took an average of 28.1 s (SD = 22.9, range: 10–75 s).

The unenhanced RAG system answered in a few seconds (4.0, SD = 1.12), while the enhanced RAG system required more time (19.1 s, SD = 4.57) due to its iterative process. A sample size calculation based on this difference (paired t-test, power = 0.8, $\alpha = 0.05$) indicated the need for 53 queries. Ultimately, 75 queries were used. These queries were selected to represent a range of potential use cases and included a mix of clinical questions encountered by radiologists, generated questions of varying difficulty, specificity, and relevance, as well as nonsensical questions designed to test the robustness of the system.

Each system's outputs were assessed on three dimensions: factual correctness (binary), completeness (5-point Likert scale), and helpfulness (5-point Likert scale), akin to the peer-reviewed process used by Truhn et al. [14]:

Question	**Rating**
The answer is complete.	Strongly disagree [1] Disagree [2] Neutral [3] Agree [4] Strongly Agree [5]
The answer is clinically useful.	Strongly disagree [1] Disagree [2] Neutral [3] Agree [4] Strongly Agree [5]
The answer is factually correct.	[0] No, [1] Yes

Table 1: Evaluation Metrics for Answer Quality

Furthermore, the answers of the enhanced and unenhanced versions were compared and rated to determine if the enhanced answer was an improvement over the unenhanced answer. The reasons for failure were noted in failure cases.

4 Results

A pilot study involving 3 radiologists with between 2.5 to X years of experience was performed with manual searches of the ESUR guidelines for 13 questions that took an average of 28.1 s (SD = 22.9, range: 10–75 s).

The unenhanced RAG system answered in a few seconds (4.0, SD = 1.12), while the enhanced RAG

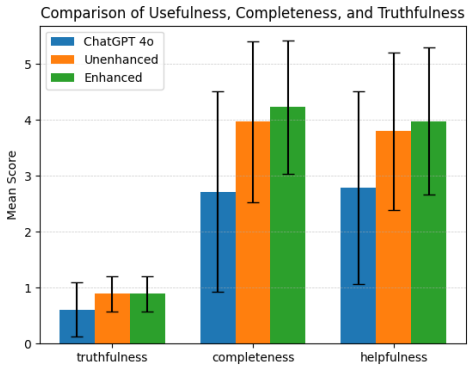

Figure 3: Truthfulness, Completeness and Helpfulness for the 3 evaluated models

system required more time (19.1 s, SD = 4.57) due to its iterative process. A sample size calculation based on this difference (paired t-test, power = 0.8, $\alpha = 0.05$) indicated the need for 53 queries. Ultimately, 75 queries were used. These queries were selected to represent a range of potential use cases and included a mix of clinical questions encountered by radiologists, generated questions of varying difficulty, specificity, and relevance, as well as nonsensical questions designed to test the robustness of the system.

The results of the questionnaire are displayed in Table 2 and visualised in Figure 3.

Statistical analysis was done in Python v3.11.8, with SciPy v1.9.3 and NumPy v1.24. To evaluate differences in performance metrics (truthfulness, completeness, and usefulness) between the Enhanced, Unenhanced, and GPT4o systems, paired-sample t-tests were employed for completeness and usefulness. For truthfulness, which involved binary ratings, the McNemar test was applied. Effect sizes for paired t-tests were quantified using Cohen's d, calculated as the mean difference divided by the pooled standard deviation. A threshold of $p < 0.05$ was considered statistically significant for all comparisons.

The statistical comparisons are summarized in Table 3. Both RAG systems significantly outperformed GPT4o across all metrics, although the differences between the enhanced and unenhanced RAG systems were modest and not statistically significant.

Model/Metric	**Truthfulness**	**Completeness**	**Helpfulness**
GPT 4o	0.61 ± 0.49	2.72 ± 1.79	2.79 ± 1.73
Unenhanced	0.84 ± 0.31	3.97 ± 1.44	3.80 ± 1.41
Enhanced	0.88 ± 0.31	4.23 ± 1.19	3.98 ± 1.32

Table 2: Performance Metrics Across Models

Models/Metrics	**T-Statistic**	**p-value**
Enhanced/GPT4o (Truthfulness)	5.61	< 0.0001
Enhanced/GPT4o (Completeness)	5.75	< 0.0001
Enhanced/GPT4o (Helpfulness)	4.86	< 0.0001
Unenhanced/GPT4o (Truthfulness)	4.04	< 0.0001
Unenhanced/GPT4o (Completeness)	5.59	< 0.0001
Unenhanced/GPT4o (Helpfulness)	5.06	< 0.0001
Unenhanced/Enhanced (Truthfulness)	0.21	0.83
Unenhanced/Enhanced (Completeness)	-0.69	0.49
Unenhanced/Enhanced (Helpfulness)	0.24	0.81

Table 3: Statistical Comparisons Between Models

5 Discussion

The findings summarized in Tables 2 and 3 show that both retrieval-augmented generation (RAG) systems significantly outperformed the baseline LLM across all evaluated measures. Although the enhanced RAG pipeline showed modest gains relative to the unenhanced version, these improvements were not statistically significant, and the effect size was small.

Mean retrieval times were 3.7 s (SD = 1.02 s) for the unenhanced pipeline and 14.45 s (SD = 4.48 s) for the enhanced pipeline, compared to a mean of 27.71 s (SD = 23.22 s) for manual lookup. Both RAG approaches were faster than human querying ($p < 0.0001$), yielding average time savings of 23.70 s (unenhanced) and 12.91 s (enhanced). Retrieval accuracy was slightly higher for the enhanced pipeline (0.85, SD = 0.36) compared to the unenhanced one (0.77, SD = 0.42), but again this difference did not achieve statistical significance.

From a clinical perspective, erroneous answers raise

legitimate safety concerns. Should an RAG system produce incomplete or wrong outputs and should those outputs form the basis of clinical decisions, suboptimal patient outcomes or even direct harm could result. Targeted improvements, such as adopting specialized embedding models, refining the chunking process, and advanced query-reformulation strategies, may enhance reliability and retrieval accuracy.

6 Conclusion

Ultimately, our findings suggest that while RAG systems represent a step forward compared to purely parametric LLMs, continued development and validation will be essential before they can be deployed safely and effectively in clinical settings.

Acknowledgments

References

1. Claveau, V. (2020). Enhancing Retrieval Processes with Query Expansion.